



TITLE:

Deterministic Graded Pushdown Automata (情報科学の数学的理論)

AUTHOR(S):

笠井, 琢美

CITATION:

笠井, 琢美. Deterministic Graded Pushdown Automata (情報科学の数学的理論). 数理解析研究所講究録 1972, 156: 83-97

ISSUE DATE:

1972-08

URL:

<http://hdl.handle.net/2433/106858>

RIGHT:

Deterministic Graded Pushdown Automata

京大・数理解析研 笠井 琢 美

序. Tree automaton と graded language

Tree automaton [4, 18, 19] とは *string* のかわりに *tree* というグラフそのものを入力とするようなオートマトンであり, それの受理する *tree* の集合は *context-free grammar* の *derivation tree* の集合と密接な関係がつかう. また有限オートマトンの自然な拡張となっており, 有限オートマトンでの結果が大部分 *tree* に関しても成立する.

よく知られているように Σ 上の *tree* t は *polish notation* ψ によって Σ 上の *string* $\psi(t)$ として表わされる.

ここでは *tree* に ψ が 1 対 1 となるような制限 (定義 1-3) をする. ここで述べる *graded language* [*] とは次のような言語のことである.

★ *tree* の集合 T が *tree automaton* に受理される必要十分条件は $\psi(T)$ が *graded language* となることである.

本稿では *graded language* が *deterministic language* [9] であることを示す.

1. Polish Notation

定義 1.1 graded alphabet とは組 (Σ, σ) のことをいう。ここで Σ は空でない有限集合で、 σ は写像 $\sigma: \Sigma \rightarrow \{0, 1, 2, \dots\}$ である。 Σ の元 a に対し $\sigma(a)$ のことを a の 階数 (rank) という。 σ の与えかたについて誤解の恐れがない場合は、 $\text{graded alphabet } (\Sigma, \sigma)$ を単に graded alphabet Σ という。以後 Σ とは graded alphabet を示すものとする。

定義 1.2 J を自然数の集合、 J^* を J によって生成される free monoid とし、 0 を J^* の単位元、 \cdot を operation とする。このとき次の (i), (ii) を満たす J^* の有限部分集合 D のことを tree domain という。

$$(i) \quad p \cdot q \in D \quad \text{なら} \quad p \in D$$

$$(ii) \quad p \cdot i \in D \quad \text{で} \quad 1 \leq i \leq J \quad \text{なら} \quad p \cdot j \in D$$

D の元を 頂点 といい、 p と $p \cdot i$ が D の元なら $p \cdot i$ を p の 子 という。(i) より $D \neq \emptyset$ なら $0 \in D$ である。頂点 0 を D の 根 という。

定義 1.3 Σ 上の tree とは写像 $\gamma: D \rightarrow \Sigma$ のことをいう。ただし D は tree domain であり、各頂点 p に対し

$$\sigma(\gamma(p)) = \max \{ i \mid p \cdot i \in D \}$$

$$= \text{頂点 } p \text{ の子の数}$$

とする。 $\gamma(p)$ を頂点 p の 名称 (label) という。

すなわち ここでいう tree γ においては、各頂点 p から出て
いる枝の数は p の名称 $\gamma(p)$ の階数と等しくなければならない。

γ の domain D を $\text{dom}(\gamma)$ で示し、 \mathcal{T}_Σ で Σ 上の tree 全体か
らなる集合を示す。 γ を $\text{graph } \{(p, \gamma(p)) \mid p \in \text{dom}(\gamma)\}$
で表わすこともある。

定義 1.4 γ を Σ 上の tree, p を γ の頂点 (すなわち $p \in \text{dom}(\gamma)$) とするとき

$$\gamma/p = \{(\gamma, a) \mid (p, \gamma, a) \in \gamma\}$$

と定める。 γ/p を頂点 p からの γ の subtree という。

定義 1.5 polish notation とは次のように定義される写
像 $\psi: \mathcal{T}_\Sigma \rightarrow \Sigma^*$ のことをいう。

$$\psi(\gamma) = \gamma(0) \psi(\gamma/1) \psi(\gamma/2) \cdots \psi(\gamma/\sigma(\gamma(0)))$$

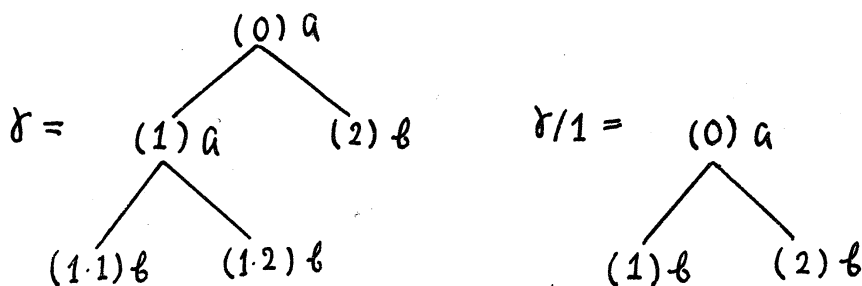
例 $\Sigma = \{a, b\}$, $\sigma(a) = 2$, $\sigma(b) = 0$

$$\gamma = \{(0, a), (1, a), (2, b), (1.1, b), (1.2, b)\}$$

とすると, γ は Σ 上の tree で 図1 のように図示される。また

$$\gamma/1 = \{(0, a), (1, b), (2, b)\}, \quad \gamma/2 = \gamma/1.1 = \gamma/1.2 = \{(0, b)\}$$

となる。



- 図1 -

$$\begin{aligned}\psi(r) &= \psi \left(\begin{array}{c} a \\ \wedge \\ b \end{array} \right) = a \psi \left(\begin{array}{c} a \\ \wedge \\ b \end{array} \right) \psi(\cdot b) \\ &= a a \psi(\cdot b) \psi(\cdot b) b = a a b b b\end{aligned}$$

定義 1.6 Σ 上の parenthesis-free formula (PFF と略す) の集合 Σ^T を帰納的に次のように定義する.

- (i) $a \in \Sigma$, $\sigma(a) = 0$ ならば $a \in \Sigma^T$
- (ii) $a \in \Sigma$, $\sigma(a) = n > 0$, $x_1, x_2, \dots, x_n \in \Sigma^T$ ならば $a x_1 x_2 \dots x_n \in \Sigma^T$

定義より 次を得る

補助定理 1. $a_1 a_2 \dots a_m \in \Sigma^*$ が PFF である必要十分条件は次の (i) と (ii) を満たすことである.

- (i) $\sigma(a_1) + \sigma(a_2) + \dots + \sigma(a_m) = m - 1$
- (ii) $1 \leq i < m$ に対し $\sigma(a_1) + \sigma(a_2) + \dots + \sigma(a_i) \geq i$

補助定理 2. (1) $\psi(\mathcal{T}_\Sigma) = \Sigma^T$ (2) ψ は one to one したがって $\psi: \mathcal{T}_\Sigma \rightarrow \Sigma^T$ は bijection である.

証明 (1) は明らか. (2) は補助定理 1 より, $a x_1 x_2 \dots x_n = a y_1 y_2 \dots y_m \in \Sigma^T$ で $x_i, y_j \in \Sigma^T$ ならば $n = m = \sigma(a)$, $x_i = y_i$, $1 \leq i \leq \sigma(a)$ となることより証明される.

補助定理 2 によつて tree r と PFF $\psi(r)$ とを同一視することができる.

2. Graded Pushdown Automaton

graded language は文法的モデル graded grammar [*] によって定義される言語である. ここでは同値なオートマトンのモデルとして graded pushdown automaton について述べる. したがって本稿においては graded language を graded pushdown automaton に受理される言語として定義する.

定義 2.1 graded pushdown automaton (gpda と略す)

とは $M = (K, \Sigma, \Gamma, \delta, Z_0, q_0, F)$ のことをいう. ここで

- (1) K は有限集合で 状態の集合, $q_0 \in K$, $F \subset K$
- (2) Σ は graded alphabet
- (3) Γ は pushdown symbol の集合, $Z_0 \in \Gamma$
- (4) $\delta: K \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \rightarrow 2^{K \times \Gamma^*}$ で 任意の (p, a, Z) $\in K \times (\Sigma \cup \{\varepsilon\}) \times \Gamma$ に対し

$$\delta(p, a, Z) \subseteq K \times \Gamma^{\sigma(a)}$$

とする. ただし ε の階数は 1 とする.

$(p, y) \in \delta(q, a, Z)$ のとき $\forall w \in \Sigma^*$, $\forall \alpha \in \Gamma^*$ に対し

$$(q, aw, Z\alpha) \vdash (q, w, y\alpha)$$

と書く. \vdash^* を \vdash の reflexive transitive closure とする.

また図示する関係上 $(p, w, \alpha) \vdash^* (q, \varepsilon, \beta)$ を $(p, \alpha) \xrightarrow{\alpha} (q, \beta)$

と書くこともある.

$$T(M) = \{ \omega \in \Sigma^* \mid (q_0, \omega, Z_0) \vdash^* (q, \varepsilon, \varepsilon), q \in F \}$$

を M によって受理される言語という。定義より $T(M) \subset \Sigma^+$ となる。一般に、ある $gpda$ によって受理される言語を graded language という。

tree automaton との関係は次のようになる。

〈オートマトン〉		〈文法〉		
tree automaton	=	regular system [5]	\mathcal{T}_Σ	
$\Downarrow \Psi$		$\Downarrow \Psi$		
gpda	=	graded grammar	Σ^+	

定義 2.2. $gpda$ $M = (K, \Sigma, \Gamma, \delta, Z_0, q_0, F)$ が ε -free であるとは $\forall p \in K, \forall Z \in \Gamma$ に対し $\delta(p, \varepsilon, Z) = \emptyset$ となることをいう。また M が deterministic であるとは ε -free で $\forall (p, a, Z) \in K \times \Sigma \times \Gamma$ に対し $\#(\delta(p, a, Z)) = 1$ となることを言う。ここで A を集合とするとき $\#(A)$ は A の元の個数を示す。

普通の pushdown automaton と同様の方法によって次の補助定理が成立する。

補助定理 2.1 任意の $gpda$ M に対し $T(M) = T(M')$ となる ε -free で 1つの state からなる $gpda$ M' が存在する。

3. The Main Theorem

定理 3.1 任意の gpda M に対し, $T(M) = T(M')$ となる deterministic な gpda M' が存在する.

証明の outline. 補助定理 2.1 より M は 1 つの state からなり ε -free であると仮定できる.

$$M = (\{p\}, \Sigma, \Gamma, \delta, Z_1, p, \{p\})$$

$$\Gamma = \{Z_1, Z_2, \dots, Z_n\}$$

$$m = \max \{ \#(\delta(p, a, Z)) \mid (p, a, Z) \in K \times \Sigma \times \Gamma \}$$

とする. Σ の各元 a と各 $Z_i, 1 \leq i \leq n$, に対し $\delta(p, a, Z_i)$ を任意に "順序" 付けてなす. j 番目の元を

$$(p, Z_{g(i,j,1)} Z_{g(i,j,2)} \dots Z_{g(i,j,k)})$$

と書く. 定義より $k = \sigma(a)$ である.

次に例でもって deterministic gpda M' の構成法を示す

$$\Sigma = \{a, b\}, \sigma(a) = 2, \sigma(b) = 0, \Gamma = \{Z_1, Z_2, Z_3\}$$

$$\delta(p, a, Z_1) = \{Z_1 Z_3, Z_2 Z_1\}$$

$$\delta(p, a, Z_2) = \{Z_2 Z_2\}$$

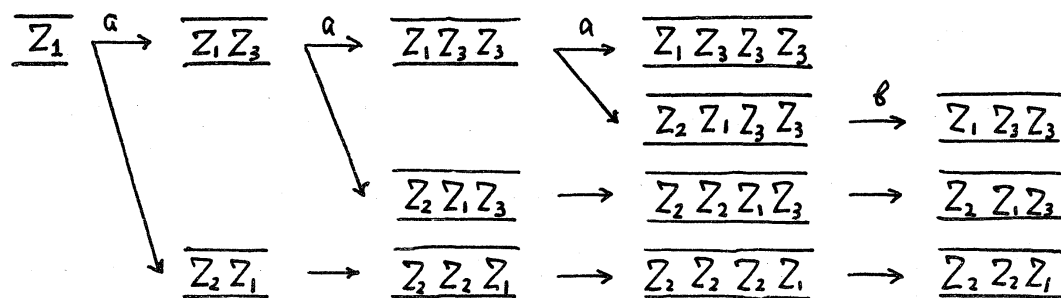
$$\delta(p, a, Z_3) = \{Z_3 Z_1, Z_3 Z_3\}$$

$$\delta(p, b, Z_1) = \phi$$

$$\delta(p, b, Z_2) = \{(p, \varepsilon)\}$$

$$\delta(p, b, Z_3) = \{(p, \varepsilon)\}$$

今, 入力 $aaaa$ を読んだ時の M のスタックの動きは



と4通りの動きが可能である。gpda の性質より、一般に M が入力 w を読んだとき可能なスタックの長さ (スタックに書かれている文字の個数) はすべて等しいことに注意する。

M を deterministic な M' でまねるためには、 M での可能なスタックすべてを M' では1本のスタックで表現しなければならない。さらに M' も gpda とするためには同じ長さのスタックで表わす必要がある。ここで例の M で入力 $aaaa$ を読んだ後の4本の長さ4のスタックを一本の長さ4のスタックで表わすことを考えよう。たてに並んだ4文字 $Z_{i_1}, Z_{i_2}, Z_{i_3}, Z_{i_4}$ を1文字 $\langle Z_{i_1}, Z_{i_2}, Z_{i_3}, Z_{i_4} \rangle$ と考えて M' での stack symbol とすることはできない。というのはたとえば例の M で入力 a^k を読んだときの可能なスタックの数は $k+1$ 本で、一般に可能なスタックの数は found されないからである。したがって次のような特別の工夫をする。各整数 $k \geq 1$ に対し、集合 $\{1, 2, \dots, k\}$ を $[k]$ で示す。 M' での stack symbol を $\langle \Sigma, L \rangle$ の型のものとする。ここで

$$\Sigma: [n] \times [m] \rightarrow \Gamma \cup \{\#\},$$

$$\cup C[n] \times [m] \times [n] \times [m]$$

である。ここの例では $n=3$, $m=2$ であるので $\langle \xi, \nu \rangle$ を

図2のように図示する。ここで上部 i 行 j 列の

まずには $\xi(i, j)$ の文字が書きこまれるものと

する。 M' の state は $[n] \times [m]$ の部分集合と

する。さて今、例に対する M' が構成されたも

のとして説明する。 M' の初期状態 q_0 とスタックの初期記号 Z_0 。

は図3上のように定める。ここで入力 aaa を入れると図3下

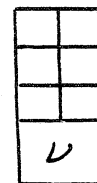


図2. M' の stack symbol

$$(q_0, Z_0) = \left(\left\{ \begin{array}{l} \langle 1, 1 \rangle \\ \langle 1, 2 \rangle \\ \langle 2, 1 \rangle \\ \langle 2, 2 \rangle \\ \langle 3, 1 \rangle \\ \langle 3, 2 \rangle \end{array} \right\}, \begin{array}{|c|c|} \hline Z_1 & \# \\ \hline \# & \# \\ \hline \# & \# \\ \hline \langle 1, 1, 1, 1 \rangle \\ \hline \end{array} \right) \xrightarrow{aaa}$$

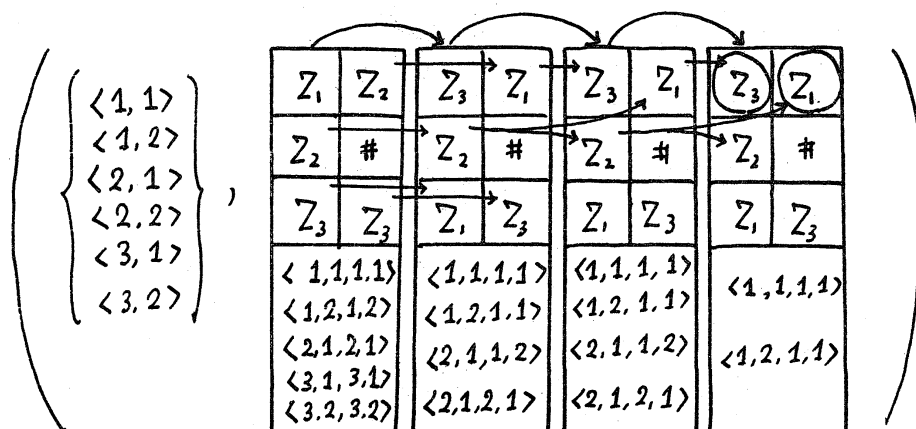


図3

ここで説明の都合上 track sequence なるものを定義する

$$(\xi_0, Z_0) \xrightarrow{\omega} (\xi, \langle \xi_1, \nu_1 \rangle \langle \xi_2, \nu_2 \rangle \cdots \langle \xi_n, \nu_n \rangle) = \theta \quad \text{in } M'$$

とするとき、次のような $[n] \times [m]$ の列 $\langle i_1, j_1 \rangle, \langle i_2, j_2 \rangle, \dots, \langle i_n, j_n \rangle$ を θ の track sequence とよぶ。

- (i) $\langle i_1, j_1 \rangle \in \delta$
- (ii) $\langle i_t, j_t, i_{t+1}, j_{t+1} \rangle \in \nu_t, \quad 1 \leq t < n$
- (iii) $\langle i_n, j_n, 1, 1 \rangle \in \nu_n$

track sequence $\langle i_1, j_1 \rangle, \dots, \langle i_n, j_n \rangle$ の $\langle i_t, j_t \rangle$ は t 番目の stack symbol の上部 i_t 行 j_t 列を指す。したがって $\langle i_t, j_t, i_{t+1}, j_{t+1} \rangle \in \nu_t$ は t 番目の座標 $\langle i_t, j_t \rangle$ は $t+1$ 番目の座標 $\langle i_{t+1}, j_{t+1} \rangle$ と結びつくことを表わす。図 3 下では、右端のまるで囲まれた座標にいたるまでの矢じるして結ばれた座標の列で、次の 4 つである。

$$l_1: \langle 1, 1 \rangle, \langle 1, 1 \rangle, \langle 1, 1 \rangle, \langle 1, 1 \rangle \quad l_2: \langle 1, 2 \rangle, \langle 1, 2 \rangle, \langle 1, 1 \rangle, \langle 1, 1 \rangle$$

$$l_3: \langle 2, 1 \rangle, \langle 2, 1 \rangle, \langle 1, 2 \rangle, \langle 1, 1 \rangle \quad l_4: \langle 2, 1 \rangle, \langle 2, 1 \rangle, \langle 2, 1 \rangle, \langle 1, 2 \rangle$$

またこの座標に書かれた記号を順にたどると、 l_1 に対しては $Z_1 Z_3 Z_3 Z_3$, l_2 に対して $Z_2 Z_1 Z_3 Z_3$, l_3 は $Z_2 Z_2 Z_1 Z_3$, l_4 は $Z_2 Z_2 Z_2 Z_1$ となる。

一般に M' は次を満足するように構成される

$$(\xi_0, Z_0) \xrightarrow{\omega} (\xi, \langle \xi_1, \nu_1 \rangle \cdots \langle \xi_n, \nu_n \rangle) = \theta \quad \text{in } M'$$

$$\langle i_1, j_1 \rangle, \dots, \langle i_n, j_n \rangle \text{ が } \theta \text{ の track sequence}$$

である必要十分条件は

$$(p, Z_i) \xrightarrow{\omega} (p, \xi_1(i_1, j_1) \xi_2(i_2, j_2) \cdots \xi_k(i_k, j_k)) \text{ in } M$$

ここで M' の構成にもどる. あと受理状態の集合 F と, 次の状態とスタックの様相を定める関数 δ' を定義すれば M' の構成は終わる. $F = \{q_0\}$, q_0 は初期状態 (図 3 上) とする.

さて, q を M' の任意の state, $\langle \xi, \nu \rangle$ を任意の stack symbol, a を Σ の任意の元とし, $\sigma(a) = k$ とするとき $\delta'(p, a, \langle \xi, \nu \rangle)$ を次のように定める.

(I) $\sigma(a) = k > 0$ のとき

$$\delta'(q, a, \langle \xi, \nu \rangle) = (q', \langle \xi_1, \nu_1 \rangle \langle \xi_2, \nu_2 \rangle \cdots \langle \xi_k, \nu_k \rangle)$$

とする. ここで $q' = [n] \times [m]$. $\xi_1, \xi_2, \dots, \xi_k$ と $\nu_1, \nu_2, \dots, \nu_{k-1}$ は a だけに depend して次のように定める.

$1 \leq i \leq n$, $1 \leq j \leq \#(\delta(p, a, Z_i))$ に対し $\delta(p, a, Z_i)$ の j 番目の元は $(p, Z_{g(i,j,1)} \cdots Z_{g(i,j,k)})$ であつた. このような (i, j) に対し $\xi_t(i, j) = Z_{g(i,j,t)}$, $1 \leq t \leq k$, と定め $(i, j, i, j) \in \nu_t$, $1 \leq t \leq k-1$, とする. すなわち $\xi_1, \xi_2, \dots, \xi_k$ から (i, j) 座標を取り出して並べると $\delta(p, a, Z_i)$ の j 番目の元となるようにする. その他の (i, j) に対しては $\xi_t(i, j) = \#$, $1 \leq t \leq k$ とする. 最後の ν_k はこの規則が適用される直前の track sequence との link の働きをするもので,

$$\begin{aligned} \nu_k = \{ \langle i', j', i'', j'' \rangle \mid \exists \langle i, j \rangle \text{ such that } \langle i, j \rangle \in \delta, \\ \langle i, j, i'', j'' \rangle \in \nu, \xi(i, j) = Z_{i'}, 1 \leq j' \leq \#(\delta'(p, a, Z_{i'})) \} \end{aligned}$$

と定められる. これは $\langle i, j \rangle, \langle i'', j'' \rangle, \dots$ なる track sequence があって $\xi(i, j) = Z_{i'}$ となるとき. a を読みは $1 \leq j' \leq \#(\delta'(P, a, Z_{i'}))$ なる各 j' に対し

$$\underbrace{\langle i', j' \rangle, \langle i', j' \rangle, \dots, \langle i', j' \rangle}_{k+1}, \langle i'', j'' \rangle, \dots$$

もまた track sequence となることを意味する.

(II) $\sigma(a) = k = 0$ のとき. $k=0$ はスタックの左端の文字を消すことを意味する. したがって track sequence $\langle i, j \rangle, \langle i', j' \rangle, \dots$ のうち $\delta(P, a, \xi(i, j)) \neq \emptyset$ となる $\langle i', j' \rangle$ を取り出せばよいかから, $\delta(q, a, \langle \xi, \nu \rangle) = (q', \varepsilon)$

$q' = \bigcup_{\langle i, j \rangle \in \xi} \{ \langle i', j' \rangle \mid \delta(P, a, \xi(i, j)) \neq \emptyset, \langle i, j, i', j' \rangle \in \nu \}$ とすればよい.

このように構成したものが $T(M) = T(M')$ となることは容易に示すことができるであろう.

系 3.1 Parenthesis language は deterministic language である.

系 3.2 $L \subset \Sigma^T$ を graded language とすれば (i) $\Sigma^* - L$ は deterministic language (ii) $\Sigma^T - L$ は graded language

系 3.3 L_1 と L_2 が graded language なら (i) $L_1 \cup L_2$ (ii) $L_1 \cap L_2$ (iii) $L_1 - L_2$ もまた graded language である.

ACKNOWLEDGMENT

The author wishes to thank Professor Satoru Takasu and Mr. Teruyasu Nishizawa for their encouragement and their helpful suggestions. The author is also indebted to Professor Hiroshi Noguchi of Waseda University for his advice and suggestions toward this paper.

REFERENCES

1. E. Altman and R. Banerji, Some problems of finite representability, *Information and Control* 8(1965), 251-263
2. M. A. Arbib and Y. Giv'e'on, Algebra Automata I: Parallel programming as a prolegomena to the categorical approach, *Information and control* 12(1968), 331-345
3. Y. Bar-Hillel, M. Perles and E. Shamir, On formal properties of simple phrase structure grammars, *Z. Phonetik Sprachwiss. Kommunikat* 14(1961), 143-172
4. W. S. Brainerd, The minimalization of tree automata, *Information and Control* 13(1968), 484-491
5. W. S. Brainerd, Tree generating regular systems, *Information and Control* 14(1969), 217-231
6. N. Chomsky, On certain formal properties of grammars, *Information and Control* 2(1959), 137-167

7. N. Chömsky, Formal Properties of Grammars, in D. Luce, R. Bush, and E. Galanter (eds.), "Handbook of Mathematical Psychology," John Wiley & Sons, Inc., New York, 1963
8. K. Culik, On some transformations in context-free grammars and languages, Czechoslovak Math. J. 17(1967), 278-311
9. S. Ginsburg and S. Greibach, Deterministic context free languages, Information and Control 9(1966), 620-648
10. S. Ginsburg and M. A. Harrison, Bracketed context-free languages, J. Comput. System Sci. 1(1967), 1-23
11. S. Ginsburg and E. H. Spanier, Control sets on grammars, Math. Systems Theory 2(1968), 159-177
12. S. A. Greibach, A new normal-form theorem for context-free phrase structure grammars, J. Assoc. Computing Machinery 12(1965), 42-52
13. D. E. Knuth, A characterization of parenthesis languages, Information and Control 11(1967), 269-289
14. B. McNaughton, Parenthesis grammars, J. Assoc. Computing Machinery 14(1967), 490-500
15. M. C. Paull and S. H. Unger, Structural equivalence of Context-free grammars, J. Comput. System Sci. 2(1968), 427-463
16. M. Rabin and D. Scott, Finite automata and their decision problems, IBM J. Res. Develop. 3(1959), 114-125
17. C. P. Schnorr, Transformational class of grammars, Information and Control 14(1969), 252-277

18. J. W. Thatcher, Characterizing derivation trees of context-free grammars through a generalization of finite automata theory, J. Comput. System Sci. 1(1967), 317-322
19. J. W. Thatcher, Generalized² sequential machine maps, J. Comput. System Sci. 4(1970), 339-367
- *. T. KASAI, Graded Languages — A Model For The Description of Derivation Tree, Preprint RIMS-93

第3回 L.A. シンポジウム (1971) 予稿集